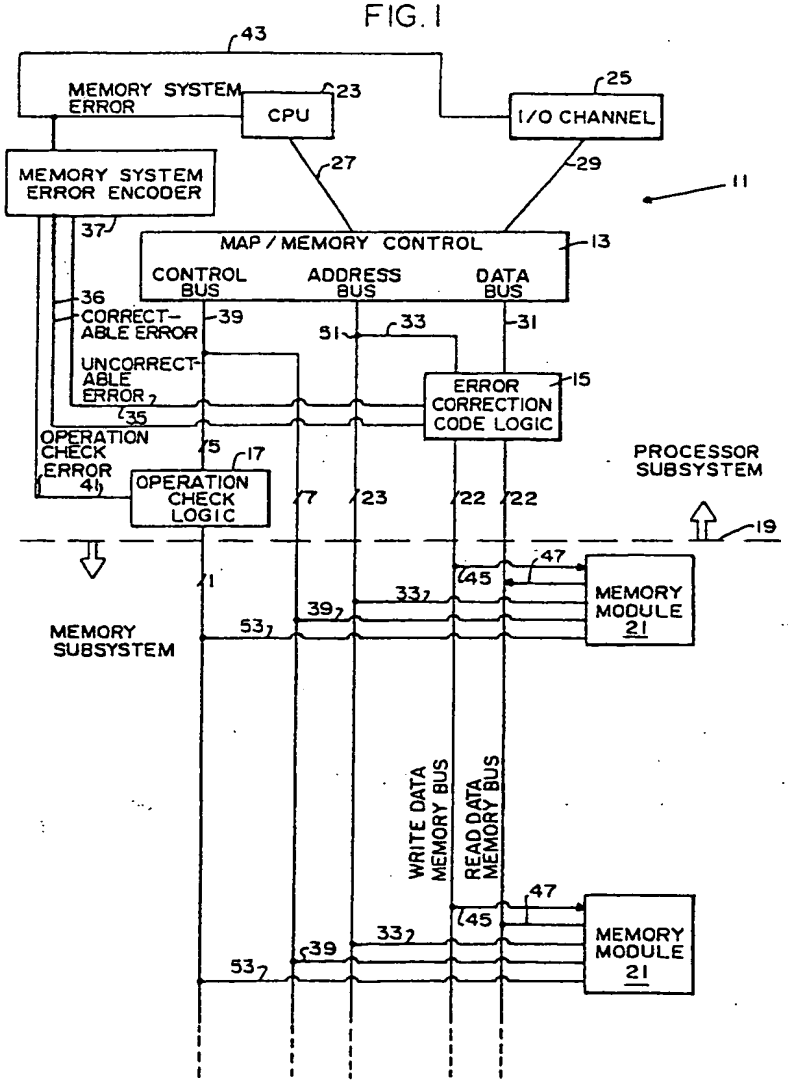
(12) UK Patent Application (19) GB (11) 2 114 335 A

(21) Application No **8301359**
(22) Date of filing **19 Jan 1983**
(30) Priority data
(31) **340611**
(32) **19 Jan 1982**
(33) **United States of America (US)**
(43) Application published **17 Aug 1983**
(51) **INT CL³ G06F 11/30**
(52) Domestic classification **G4A** 12C 12R 13E FM
(56) Documents cited
GB 1442665
GB 1433608
GB 1397274
(58) Field of search **G4A**
(71) Applicants
Tandem Computers Incorporated, (USA - California), 19333 Vallco Parkway, Cupertino, State of California 95014, United States of America.
(72) Inventors
Richard Anthony Humphrey, Steven Derek Fisher, Steven Warren Wierenga, Jon Sjostedt.
(74) Agent and/or Address for Service
W. P. Thompson and Co., Coopers Building, Church Street, Liverpool L1 3AB.

(54) Computer memory control systems

(57) A memory system (11) for a computer includes logic which detects data errors, address errors and operation errors to increase the reliability of data stored in the memory system. Address errors are detected by encoding address parity information into the data check field of each memory location. A signal is generated in each memory module (21) indicating the status of operations of that memory module and is transmitted to the processor subsystem (13, 15, 17 & 37) of the computer for comparison with a signal indicating the status of operations of the processor subsystem to ensure that all memory modules (21) and the memory control in the processor are receiving the same commands. An interrupt is provided if there is a difference between the commands.

FIG. 1



GB 2 114 335 A

FIG. 1

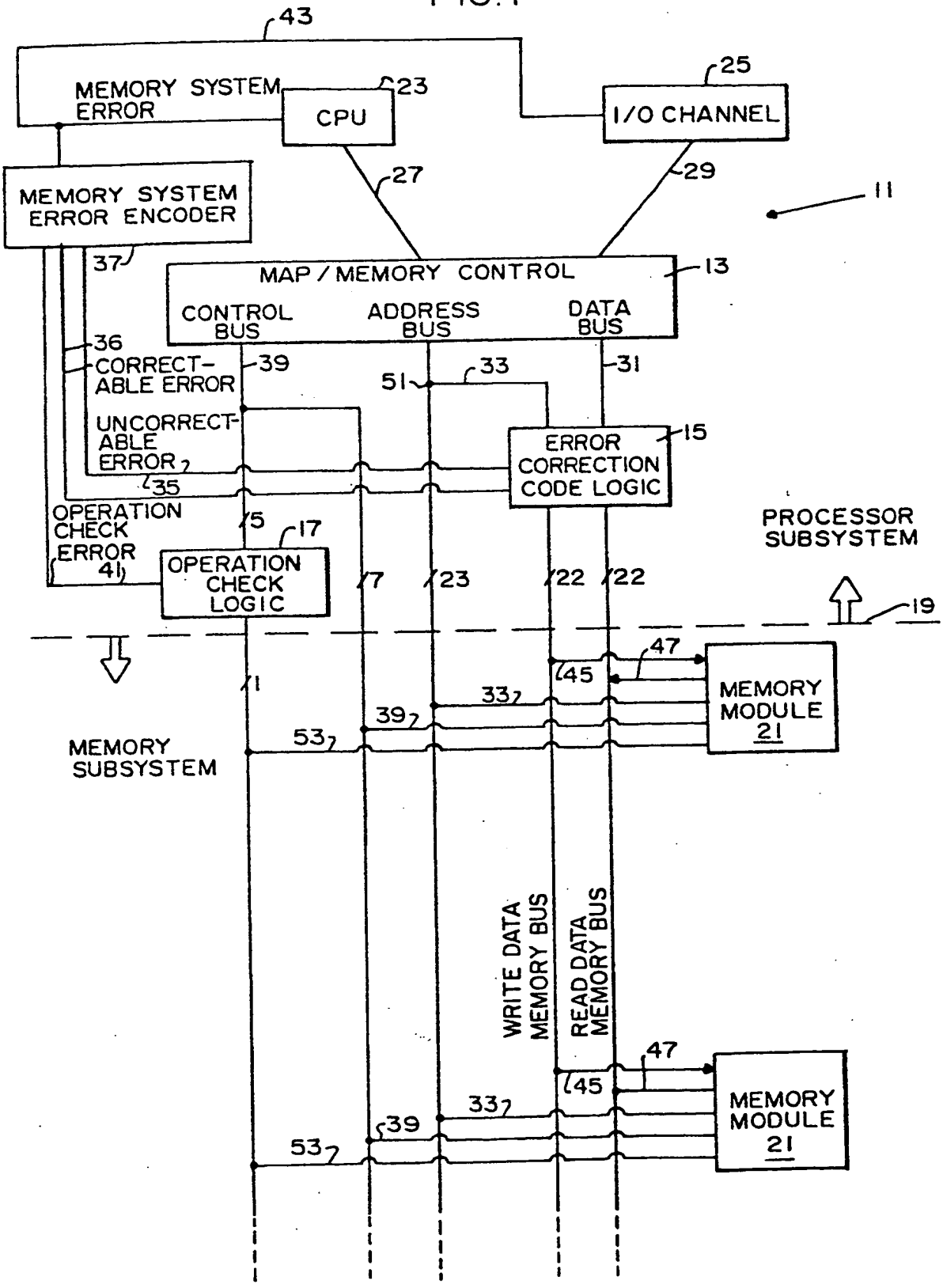

43
MEMORY SYSTEM ERROR
CPU
23
25
I/O CHANNEL
MEMORY SYSTEM ERROR ENCODER
27
29
11
37
MAP / MEMORY CONTROL
13
CONTROL BUS
ADDRESS BUS
DATA BUS
36
CORRECT-ABLE ERROR
39
51
33
31
UNCORRECT-ABLE ERROR
35
ERROR CORRECTION CODE LOGIC
15
OPERATION CHECK ERROR
41
5
17
OPERATION CHECK LOGIC
7
23
22
22
PROCESSOR SUBSYSTEM
19
47
45
MEMORY MODULE 21
1
33
39
53
MEMORY SUBSYSTEM
WRITE DATA MEMORY BUS
READ DATA MEMORY BUS
47
45
MEMORY MODULE 21
33
39
53

FIG. 2 SEMICONDUTOR MEMORY MODULE

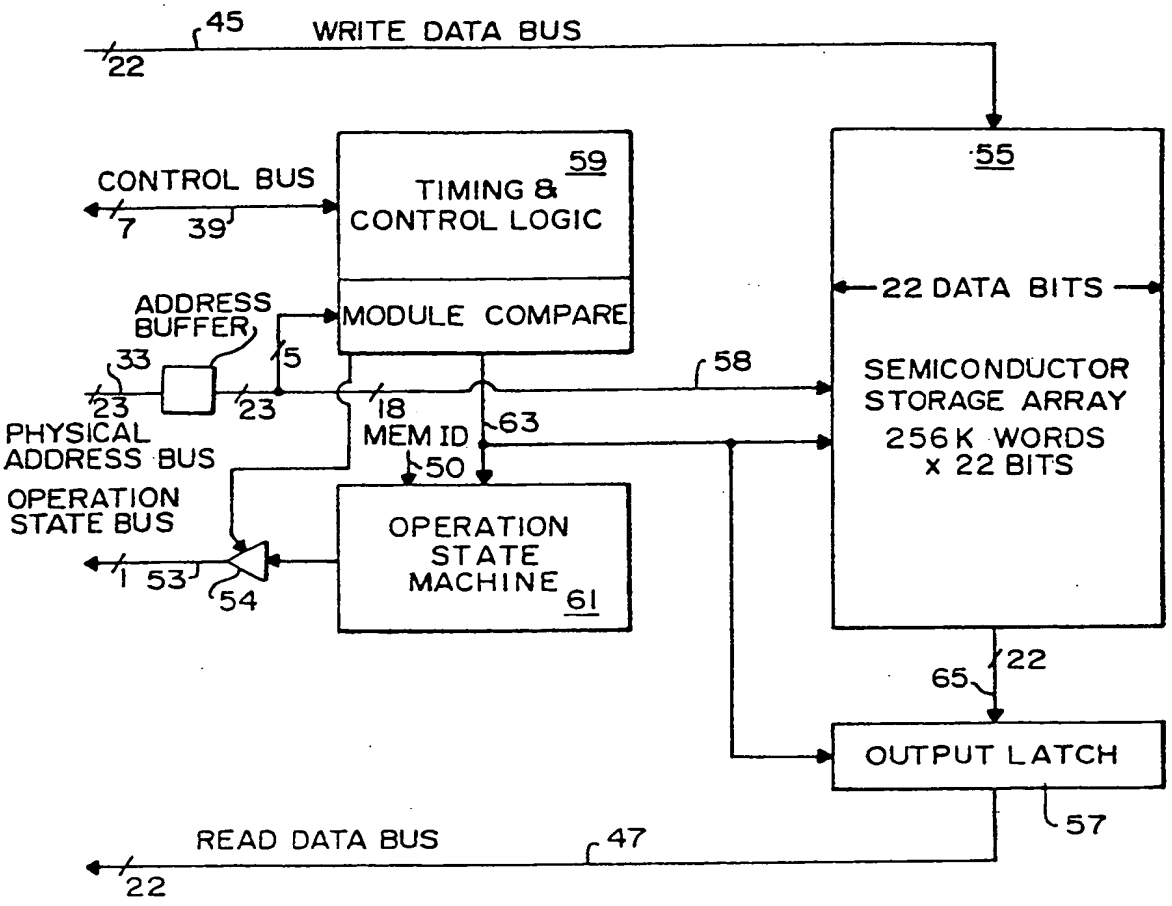
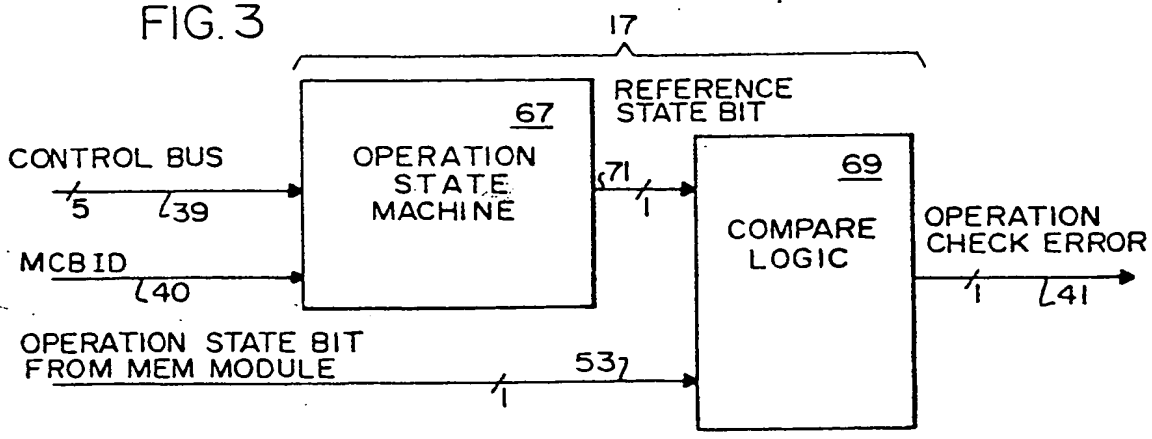


FIG. 3



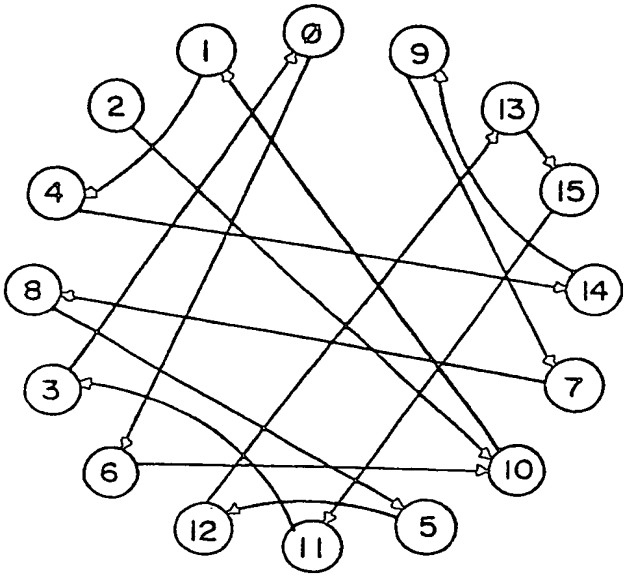


FIG. 4-A
REFRESH

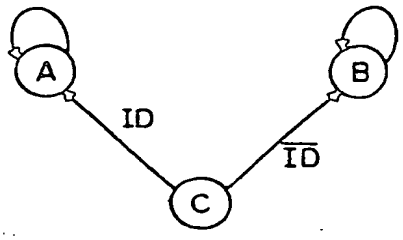
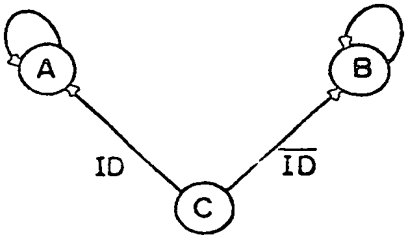
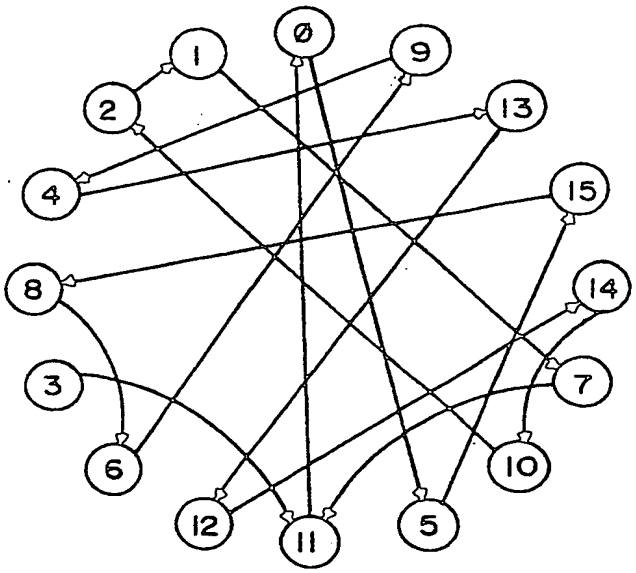


FIG. 4-B
WRITE





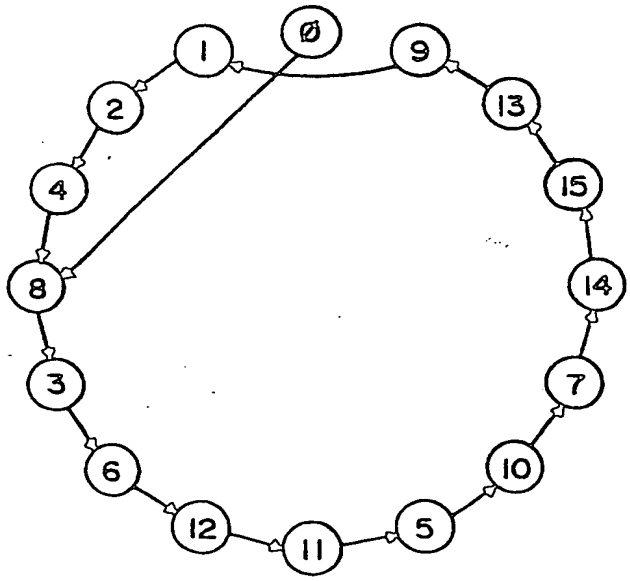
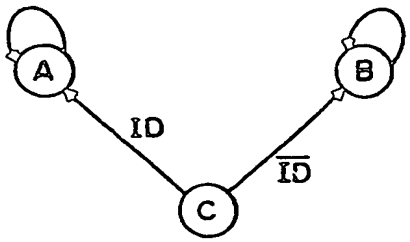


FIG. 4-C
READ



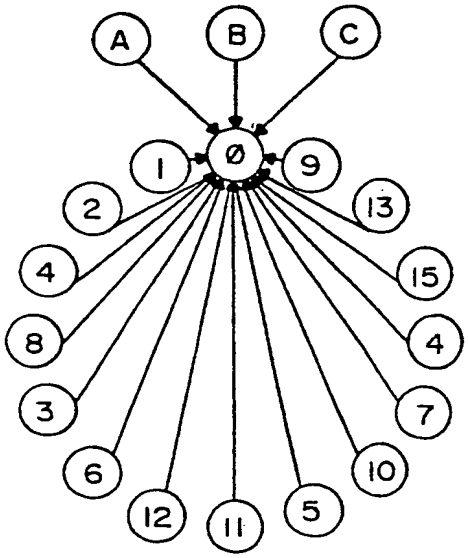
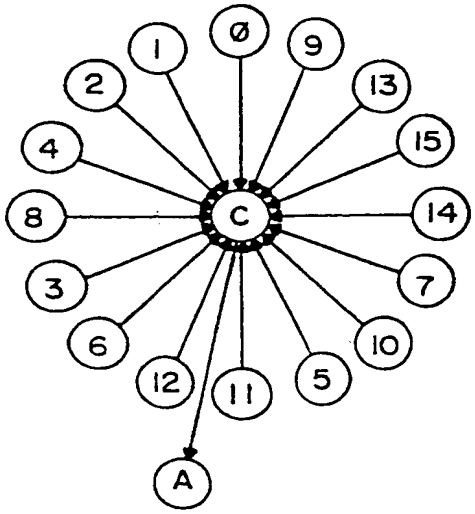


FIG. 4-D
RESET

FIG. 4-E

$$ID\left[(\overline{READ}\cdot\overline{REFRESH}\cdot\overline{WRITE})+(READ\cdot REFRESH)+(READ\cdot WRITE)+(WRITE\cdot REFRESH)\right]$$



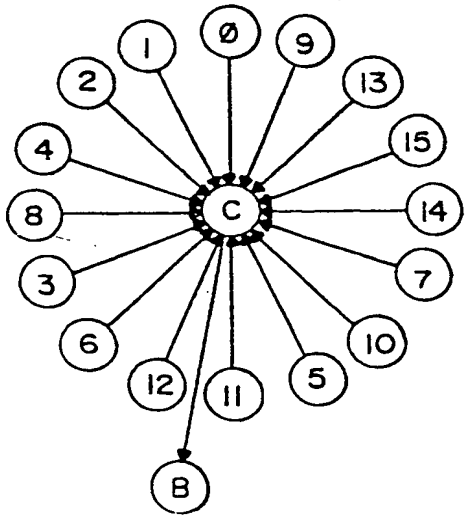
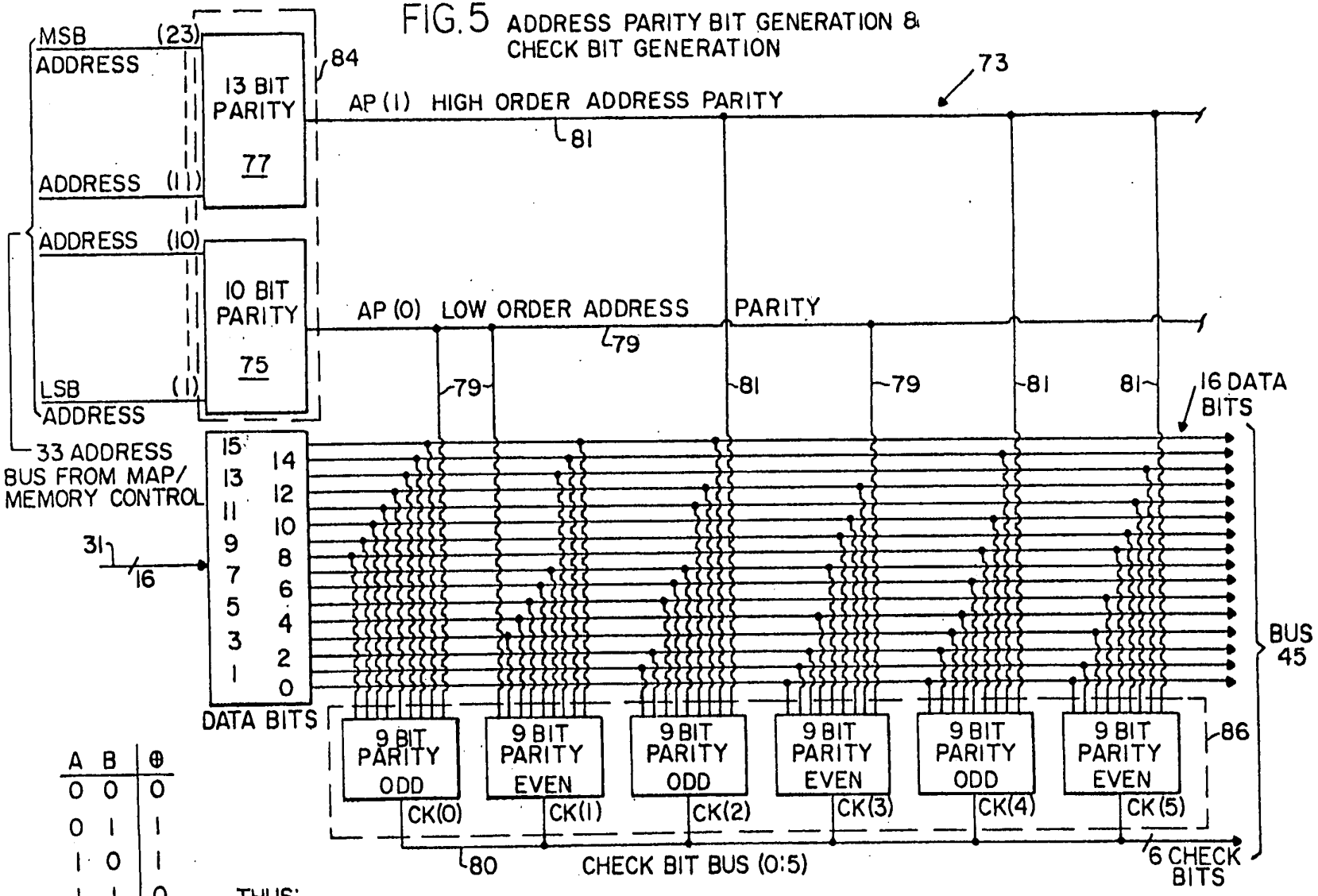


FIG. 4-F

$$\overline{ID}\left[(\overline{READ}\cdot\overline{REFRESH}\cdot\overline{WRITE})+(READ\cdot REFRESH)+(READ\cdot WRITE)+(WRITE\cdot REFRESH)\right]$$

FIG. 5 ADDRESS PARITY BIT GENERATION & CHECK BIT GENERATION



A	B	⊕
0	0	0
0	1	1
1	0	1
1	1	0

THUS:

$CK(0) = D8 \oplus D9 \oplus D10 \oplus D11 \oplus D12 \oplus D13 \oplus D14 \oplus D15 \oplus AP(0)$

$CK(3) = \overline{D0 \oplus D1 \oplus D2 \oplus D4 \oplus D7 \oplus D9 \oplus D10 \oplus D12 \oplus AP(0)}$

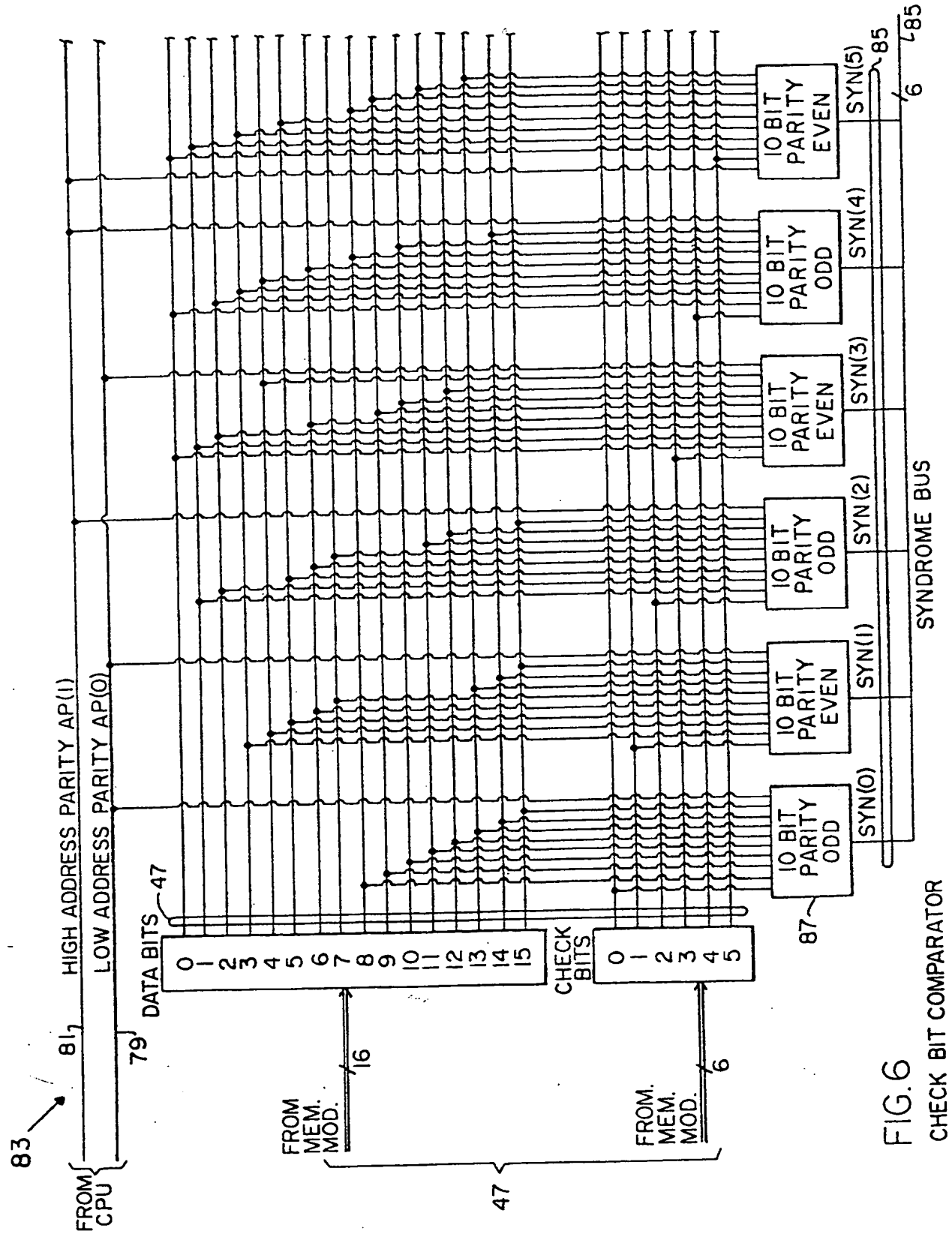


FIG.6
CHECK BIT COMPARATOR

FIG. 7

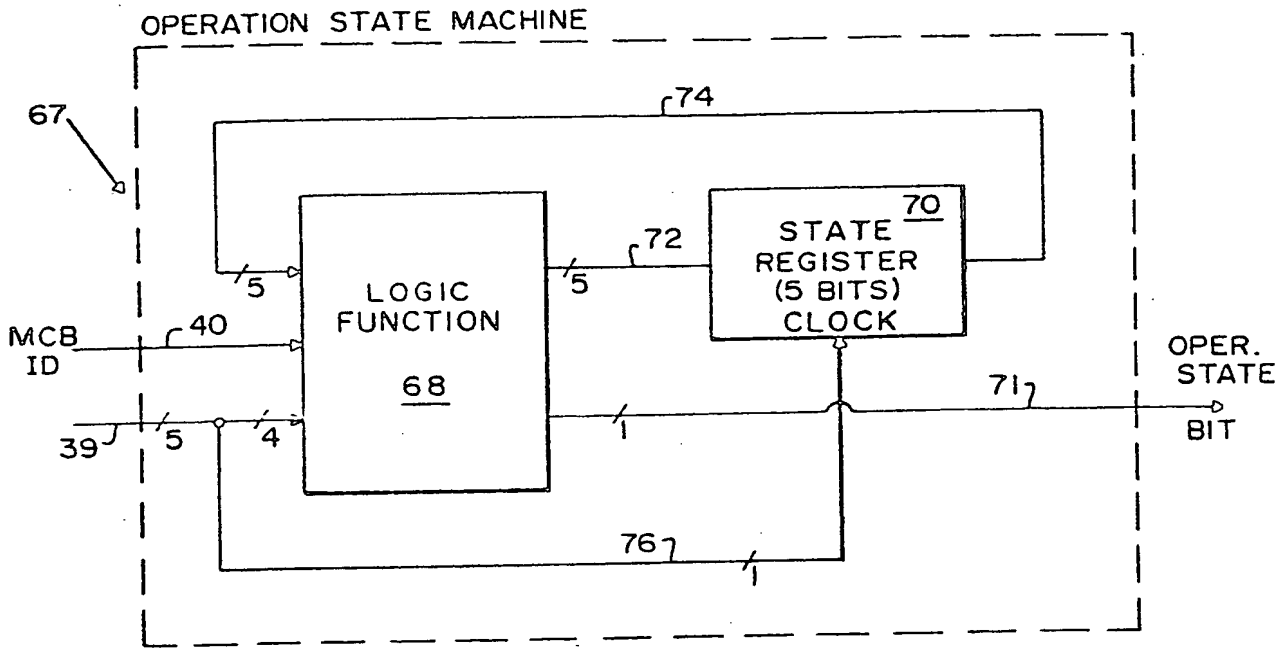
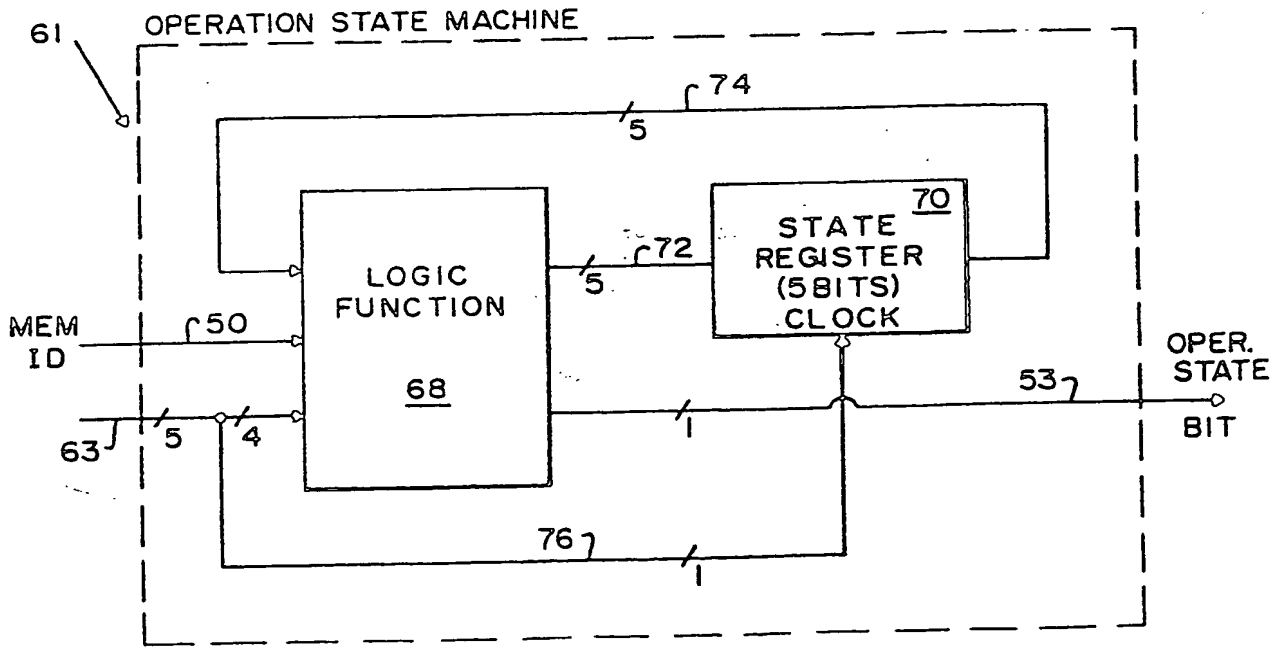

OPERATION STATE MACHINE
67
74
5
LOGIC FUNCTION
68
72
5
STATE REGISTER (5 BITS) CLOCK
70
MCB ID
40
39
5
4
1
71
OPER. STATE BIT
76
1

FIG. 8


61
OPERATION STATE MACHINE
74
5
LOGIC FUNCTION
68
72
5
STATE REGISTER (5BITS) CLOCK
70
MEM ID
50
63
5
4
1
53
OPER. STATE BIT
76
1

SPECIFICATION

Improvements in and relating to computer memory control systems

The invention relates to improvements in and relating to computer memory control systems, and more particularly concerns a word organized, random access, read-write or read-only memory of the kind having a semiconductor array with a data field and a data check field for detecting data errors in each word. The invention is applicable also to memory arrays other than semiconductor arrays, for example, core memory and other types of bit storage means. The memory word may represent more than one data word in a particular computer.

The present invention is applicable to a memory system of the kind in which a processor module is associated with one or more memory modules.

The memory system of the present invention permits data errors, addressing errors and operation errors to be detected.

Errors in stored data can be caused by the sticking of one or more bits in a data word stored in a semiconductor memory, and such errors can be detected by associating a data check field with the data field of the stored word. Various types of errors can be detected by codes associated with the data check fields. For example, with a Hamming code like that disclosed in U.S. Patent Specification No. 4,228,496 all single bit data errors can be detected and corrected, all double bit data errors can be detected and some errors involving more than two data bits can be detected.

To increase the overall reliability of a memory system, it is desirable to detect addressing errors as well as errors in the stored data.

It is known to build a self-checking memory address decoder in the straightforward way (by duplicating and comparing outputs;) but this can become prohibitive in size and cost as the size of the address increases. Thus, in a 20 bit address there are two to the twentieth power decoder outputs, requiring roughtly two to the eighteenth power (about 262,000) integrated circuits just to compare. In addition, in a real main memory the address decoding is carried out at several levels; first a "module compare" to select one of several PC boards, then a "row decode" to select one word of memory devices, then internal X-Y decoders to select an individual bit cell in the memory devices. This creates further problems in obtaining a self-checking operation of memory address.

Obtaining satisfactory address error detection with a reasonable integrated circuit package count has been a problem in the prior art.

Data error detection and/or correction itself does not protect against an operation error in a memory system. For example, if contention on a bus should result in producing a write command at the wrong time, the error system for detecting stuck bits in a data field would not give any protection against the memory error resulting from the erroneous operation command.

To provide high reliability in a memory system, it is desirable to ensure that the operations of each memory module are in step with the operations of the associated processor module.

It is a primary object of the present invention to detect address errors by coding address parity information into the data check field of each memory location.

It is a related object to detect address errors by an encoding technique that does not require storing the address for address error detection.

It is another related object to increase the system-level reliability by a substantial amount, as much as 10 to 100 times, by including the address parity bit method in an already existing data check code generation system so that address error detection can be obtained at essentially no cost.

It is a further object of the present invention to ensure that all memory modules and the memory control in the processor are receiving the same commands.

It is a related object to detect any difference in the operations sequence between the processor and all memory modules and to provide an interrupt to the processor if there is a difference.

It is a related object to generate a signal in each memory module indicating the status of operation of that memory module and to transmit that signal to operation check logic in the processor for comparison to the statues of operations of the processor module.

It is a still further object of the present invention to combine data error, address error and operation error detection in a memory system to increase the system level reliability.

In the present invention a processor module is associated with one or more memory modules in a way that permits data errors, addressing errors, and operation errors to be detected.

The memory system of the present invention includes a processor subsystem and a memory subsystem.

The processor subsystem comprises a map/memory control, error correction code logic, operation check logic and a memory system error encoder. All of these components are physically located in the processor subsystem portion of the memory system. This configuration assures that address faults arising in cabling and connectors are detected that might otherwise go undetected if the error correction code structure were located in the memory modules.

Each memory module comprises a semiconductor storage array, timing and control logic, and buses for connecting the storage array to the map/memory control and error correction code logic and for connecting the timing and control logic to the map/memory control.

Each memory module also has an operation state bus connectible to the operation check logic of the processor module and an operation state machine associated with the operation state bus and the timing and control logic for generating a signal indicating the status of operations of the memory module and for transmitting that signal to the operation check logic for comparison to the status of operations of the processor module.

In a specific embodiment of the present invention, the semiconductor storage array has sixteen data bit positions and six check bit positions forming a twenty-two bit storage word.

The error correction code is a Hamming code effective to detect and correct all single data bit errors, and to detect all double data bit errors and to detect some data bit errors that involve more than two bits.

In the present invention the address parity information is coded into the six bit check field in a way that permits the error correction code logic to detect all single bit address errors and to detect some multiple bit address errors, even in the presence of a single bit data error.

In a particular embodiment of the present invention, the encoding for address error detection is done in stages. In a first stage a 23 bit address is encoded in a 13 bit parity tree and a 10 bit parity tree to produce two parity bits on separate lines. These two parity bits are then encoded according to a connection pattern of a nine bit parity tree into the six check bit field. Thus, the 23 bit address is encoded as part of the six check bits in the 22 bit data code word supplied to the semiconductor storage array.

On a subsequent read operation, a data code word is read from the memory array and is combined with newly generated read address parity bits transmitted to a check bit comparator. In the check bit comparator these two address parity bits are combined with the data code word read to generate a syndrome on a syndrome bus that indicates the type of error, if any.

Since, as noted above, the error-correction code itself will not protect against errors resulting from improper operation, such as for example, an improper write command, the present invention incorporates an operation state mechanism in each memory module for generating a signal indicating the status of operations in that memory module. This generated signal is then compared with a signal representing the status of operations in the processor module, and the operation check logic detects any difference in the status of operations and provides an error signal resulting in an interrupt if there is any logical difference between the two signals.

Memory system apparatus and methods that incorporate the structure and techniques described above and that are effective to function as described above constitute further, specific objects of this invention.

Other and further objects of the present invention will be apparent from the following description and claims and are illustrated in the accompanying drawings which, by way of illustration, show preferred embodiments of the present invention and the principles thereof and what are now considered to be the best modes contemplated for applying these principles. Other embodiments of the invention embodying the same or equivalent principles may be used and structural changes may be made as desired by those skilled in the art, within the scope of the appended claims.

Figure 1 is a block diagram of a memory system constructed in accordance with one embodiment of the present invention.

Figure 2 shows further details of the memory module portion of the memory system shown in Figure 1.

Figure 3 is a view of the operation state machine and compare logic of the operation check logic portion of the memory system shown in Figure 1.

Figures 4A - 4F are state machine diagrams for the state machines which are incorporated in the operation check logic of the memory subsystem and in the map/memory control of the processor subsystem of Figure 1.

Figure 5 is a diagram of the address parity bits generator and check bits generator used in the error correction code logic 15 of the memory system shown in Figure 1. Figure 5 also lists logic equations for two of the 9-bit parity trees used in the check bit generator.

Figure 6 shows the check bit comparator incorporated in the error correction code portion of the memory system shown in Figure 1.

Figure 7 shows details of the operation state machine 67 of Figure 3, and

Figure 8 shows details of the operation state machine 61 of Figure 2.

Referring to the drawings, a memory system constructed in accordance with one embodiment of the present invention is indicated generally in Figure 1 by the reference numeral 11.

The memory system 11 is incorporated in a multiple processor system of the kind shown in U.S. Patent Specification No, 4,288,496.

The memory control system of the present invention has specific use in a multiple processor system of the kind shown in U.S. Patent Specification No. 4,228,496, but is not restricted to use in such systems. The memory system of the present invention also has utility in a single processor system.

The memory system of the present invention will now be described with reference to Figure 1 and Figures 2 - 8 of the drawings of this application. Figure 1 of this application corresponds generally with Figure 34 of the U.S. Patent Specification mentioned.

The memory system as shown in Figure 1 comprises a map/memory control 13, an error correction code logic 15, an operation check logic 17, and a memory system error encoder 37, all of which are physically located in the processor subsystem portion of the memory system (as indicated diagrammatically by the arrow pointing upwardly from the dashed line 19) and one or more memory modules 21 which are physically

located in the memory subsystem (as indicated diagramatically by the arrow pointing downwardly from the dashed line 19).

The map/memory control 13 is associated with a central processor unit 23 (CPU) by a bus 27 which comprises data, logical address, and control lines, and an input/output (I/O) channel 25 of the processor subsystem as illustrated in Figure 1. The I/O channel 25 is connected to the map/memory control 13 by a bus 29 which also comprises data, logical address and control lines.

In the particular embodiment illustrated in Figure 1 the central processor unit 23 corresponds in general to the CPU 105 shown in Figure 34 of U.S. Patent Specification No. 4,228,496, and the I/O channel 25 of Figure 1 corresponds generally to the I/O channel 109 of the same Specification.

The error correction code logic 15 is associated with the map/memory control 13 by a data bus 31 and an address bus 33. The error correction code logic 15 has output lines 35 and 36 connected to the memory system error encoder 37. The output line 35 signifies "uncorrectible error", while the output line 36 signifies "correctible error".

The operation check logic 17 is connected to the map/memory control 13 by a control bus 39. The operation check logic 17 is also connected to the memory system error encoder 37 by an operation check error line 41.

The memory system error encoder 37 has a memory system error output bus 43 which is, in turn, connected to the CPU 23 and input/output channel 25 as illustrated in Figure 1.

Each memory module 21 has five bus connections which associate the memory module 21 with the map/memory control 13, the error correction code logic 15 and the operation check logic 17. Thus, each memory module 21 is connected to the error correction code logic 15 by a bidirectional data bus comprising two unidirectional data buses, an input (write) data bus 45 and an output (read) data bus 47.

Each memory module 21 is connected to the map/memory control 13 by an address bus 33, interconnected to the address bus 33 from error correction code logic 15 at point 51.

Each memory module 21 has an operations state line 53 connected to the operation check logic 17.

Each memory module 21 is also connected to the map/memory control 13 by the control bus 39. The control bus 39 comprises seven lines. Only five of the lines go to the operation check logic 17, however, all seven lines of the control bus 39 are connected to each memory module 21. The number of bits on the control bus 39 and on other buses shown in the various Figures of the drawings are indicated by the small numerals associated with slash marks on the bus lines.

The memory system shown in Figure 1 includes four important functions that work in conjunction with and in addition to the basic operations of storing and retrieving data with word-addressable access, correction of single data errors, and detection of double data errors as described in the referenced patent. These functions will be described in more detail later in this specification.

The first function is address error detection which is performed by the error correction code logic 15 in combination with the memory module 21. Address error detection ensures that any faults in address transmission, comparison, and decoding logic which occur between the time of writing and reading a memory location are detected. The structure for performing this function is physically located in the processor subsystem. This configuration assures the address faults additionally arising in cabling and connectors are detected that might otherwise go undetected if that structure were located in the memory modules.

The second function is the generation of a signal in the memory module 21 to be used for operation error detection. This function is provided by operation state machine logic located in the memory module 21.

The third function is memory operation error detection which is performed by the operation check logic 17 in combination with the memory module 21.

The fourth function is the combination of the address error detection and the memory operation error detection.

Figure 2 shows more detail of the memory module 21 which comprises a semiconductor storage array 55 and an output latch 57.

The memory module 21 also includes timing and control logic 59 and operation state machine 61.

The timing and control logic 59 is connected to the operation state machine 61, the semiconductor storage array 55 and the output latch 57 by a bus 63 which comprises control signal lines.

The semiconductor storage array 55 is associated with the output latch 57 by a bus 65 which comprises data lines.

Referring again to Figure 2, the first function of a memory module 21 is a storage function; the semiconductor storage array 55 in combination with the output latch 57 and the timing and control logic 59 provide a storage function for storing and retrieving data. The three basic operations of the storage function of the memory module are read, write, and refresh.

The second function of the memory module 21, that of the operation check, employs the operation state machine 61. If (1) a missed operation, (2) a null operation – no READ, no WRITE, no REFRESH, and START, (3) a multiple operation, for example, READ and WRITE and START, or (4) an incorrect operation, for example, READ instead of WRITE, and START occurs, then the operation state machine 61 changes state in a sequence different from that of the operation state machine 67 in the operation check logic 17 of Figure 3.

This second function provides signals representing the state of the operation state machine 61 in each memory module 21 to the operation check logic 17 for checking the operation of the memory module 21. The

operation state machine 61, based on signals representing the current state of timing and control logic 59 generates a signal indicating the status of memory module timing and control signals (read, write and refresh signals in their sequence) which is transmitted on the operation state line 53 to the operation check logic 17 (see Figure 1). The signal transmitted on line 53 allows the operation check logic 17 to perform the operation check as described later in this application.

As shown in Figure 3, the operation check logic 17 comprises an operation state machine 67 and a compare logic 69. The operation state machine 67 is connected to the compare logic 69 by a line 71 for transmitting a reference state bit to the compare logic 69.

As shown in Figure 7, the operation state machine 67 comprises a logic function 68 and a state register 70. The control bus 39 supplies inputs to the logic function 68. A bus 72 connects the logic function to the state register 70, and a bus 74 feeds back signals from the state register 70 to the input of the logic function 68. A line 76 from the control bus 39 supplies a clock signal to the state register 70.

The logic function 68 implements the next state function of the state diagram in Figures 4a - 4f.

In a specific implementation, the state register comprises five flip flops for storing the current state while the logic function 68 is computing the next state.

As shown in the state diagrams in Figures 4A - 4F, the next state in any cycle depends upon the current state and the type of the current cycle. Four of the five signals on the bus 39 determine the type of cycle and the remaining signal is the clock.

The operation state machines 67 and 61 are identical. However, the operation state machine 61 is distinguished from the operation state machine 67 by the signal present on input 50 and 40, respectively. With certain cycles (null, multiple type), it is a known error condition and the state machine changes state indirectly through a state "C" (see Figures 4E and 4F) to a "lock" state which is maintained until reset. There are actually two "lock" states, A and B. If the state machine is in the operation check logic 17 (MCB ID = 1 on line 40), the stae machine makes a transition to the A state, and if it is in a memory module 21 (MEM ID = 0 on line 50), it goes to the B state. If operation state machine 67 and operation state machine 61 are both in their respective locked states, then the reference state bit and operation state bit from the memory disagree, causing an operation check error.

The compare logic 69 (see Figure 3) checks the operation state bit from the memory module on the line 53 against the reference state bit from the operation state machine 67 on the line 71. The compare logic 69 generates a signal on the line 41 which indicates whether or not the two input signals on the lines 53 and 71 are logically equal. If they are not logically equal, the output signal on the line 41 indicates that the operation state machine 67 does not agree with the operation state machine 61 in the memory module 21 (see Figure 2) and an operation error is identified. Note that any operation error is uncorrectible; thus, an operation check error will cause the memory system error lines 43 to signal that an uncorrectible memory error has occurred, via the memory system error encoder 37.

In error-free operation, the signals on the lines 53 and 71 will not be logically different. However, if they are logically different an operation check error is identified, and an error signal is produced by the memory system error encoder 37. As will be described in more detail below, the operation check for a particular memory module is performed only during a read cycle of that memory module.

The combined operation of the operation state machine 61 and the operation check logic 17 protects against a number of factors including driver failures, receiver failures and cable failures which cause incorrect memory module operations. It also protects against memory module failures which preclude any module operation.

Figures 4A - 4F are diagrams showing the states through which the operation state machine 67 of the processor subsystem and the operation state machine 61 of a memory module in the memory subsystem sequence in response to the input signals supplied to them. Figures 4A - 4F are overlays of state transitions for one state machine. They are presented separately as illustated in the specific Figures 4A, 4B, etc., for ease of understanding the transitions in particular cycles. Thus, for example, state 6 is the same for all cycles and for all of the Figures 4A - 4F.

Figure 4A shows the state transitions caused by a refresh cycle.

Figure 4B shows the state transitions for a write cycle.

Figure 4C shows the state transitions for a read cycle.

Figure 4D shows the state transitions for a reset cycle.

Figure 4E shows the state transitions for other cycles which are known error conditions, if the input signal 40 or 50 is a logical zero.

Figure 4F shows the state transitions for other cycles which are known error conditions, if the input signal 40 or 50 is a logical one.

Looking at Figure 4D, after a reset cycle all the machines are in the state zero. Looking now at Figure 4A, in response to a refresh cycle, the state machines would change state from state zero to state 6.

If continued refresh cycles are performed, then the continued state machine transitions would continue as indicated by the arrows in Figure 4A. That is, the next transition would be from state 6 to state 10, the next transition would be from state 10 to state 1, etc. With continued reference to Figure 4A, if, after starting from state zero and changing state to state 6 (as described above) then the next cycle is a write cycle (as illustrated in Figure 4B), then all the state machines will change state from state 6 to state 9.

Figures 4C - 4F may be interpreted in a manner similar to that for Figures 4A and 4B.

Figures 5 and 6 are detailed views of different parts of the error correction code logic 15 shown in Figure 1. Figure 5 shows details of address parity bit generation and check bit generation in a generator unit 73.

The 16 data bits and the six check bits are transmitted from the error correction code logic 15 to the memory modules 21 on the bus 45 (see Figure 1). It should be noted that, while 16 data bits and six check bits have been illustrated in the particular embodiment to be described, the system is applicable to either more or less of each according to the principles of single data error correcting and double error detecting Hamming codes.

The 9-bit parity trees of the check bit generator 86 shown in the lower part of Figure 5 are basically similar in structure and mode of operation to the 8-bit parity trees of the check bit generator shown in Figure 38 of U.S. Patent Specification No. 4,228,496, except that an additional input has been added to each parity tree and has been included in the logic equations. However, the truth table for the exclusive -or operation shown in Figure 5 applies to both an 8-bit and a 9-bit parity tree implementation.

The high and low order parity bits produced on lines 81 and 79 are based on the address. The address parity bit generator 84 shown in Figure 5 comprises a 10-bit parity tree 75 for address bits 1 through 10 and a 13-bit parity tree 77 for address bits 11 through 23. The parity tree 75 generates the low order address parity bit on line 79 which goes into check bit generators 0, 1, and 3. The 13-bit parity tree 77 generates the high order address parity bit on a line 81, and this line 81 supplies that bit to check bit generators 2, 4, and 5.

The generator 86 produces six check bits on 80, part of the bus 45, for storage in the memory array 55 of Figure 2. These six check bits are used in connection with the 16 data bits to detect all single and double bit memory errors and to detect some three or more bit memory errors. The six check bits are also used in conjunction with the 16 data bits to correct single data bit errors. This mode of operation is the same as that described in connection with Figure 38 of U.S. Patent Specification No. 4,228,496.

The two lines 79 and 81 comprise address parity information which is encoded into the six check bits on a write operation as described previously. During a subsequent read operation, lines 79 and 81 comprise address parity information based on the address for the current READ operation. These lines (79 and 81) are checked against the address parity information encoded in the six check bits read from the memory module and if the address parity information is found to differ, an uncorrectible memory error signal is asserted.

A typical addressing error that this invention would detect is a stuck address bit on address bus 58 (see Figure 2), or a stuck address bit within the memory module itself. A word may be written to location ∅ with address parity information for address ∅. If bus 58 had the least significant address bit stuck at ∅, then a READ operation to address 1 on bus 33 would transmit an address of ∅ on but 58. The six check bits read contain a code for address parity of ∅. The error correction code (ECC) logic 15 indicates an addressing error.

Figure 6 shows details of the check bit comparator 83 of the error correction code logic 15 shown in Figure 1.

The check bit comparator 83 has three inputs, the lines 79 and 81 (see Figure 5) and the read data bus 47. The check bit comparator 83 has an output 85 which is a syndrome bus.

In the check bit comparator 83, the 16 data bits of the read data bus 47 are connected to the parity trees as illustrated.

In the check bit comparator 83, the 16 data bits of the read data bus 47 are connected to the parity trees as illustrated, and the six check bits of the read data bus 47 are similarly connected to the parity trees as illustrated.

The parity trees 87 function in the same way as the parity trees 505 shown in Figure 39 of U.S. Patent Specification No. 4,228,496 except for the fact that the parity trees 87 are 10-bit parity trees while the parity trees 505 are 9-bit parity trees.

In accordance with the present invention, on a read cycle, lines 79 and 81 indicate the parity of the address which is being read. If this does not agree with the address parity encoded in the check bits, then a code is generated on the syndrome bus 85 to indicate an addressing error.

The check bit comparator 83 shown in Figure 6 is used only during a read cycle. The address parity bit generator 84 in Figure 5 is used during both read and write cycles. The check bit generator 86 shown in Figure 5 is used only during a write cycle.

The error correction code logic 15 shown in Figure 1 also includes a syndrome decoder which received the signal on the syndrome bus 85 (see the signals on the syndrome bus 85 in Figure 6), and the syndrome decoder is the same as the syndrome decoder shown in Figure 40 of U.S. Patent Specification No. 4,228,496.

The error correction code logic 15 also includes a bit complementer which is not shown in the drawings of this application but which performs the same function as the bit complementer shown in Figure 41 of U.S. Patent Specification No. 4,228,496.

The syndrome code which is transmitted on the syndrome bus 85 is used, in the present invention, to identify address errors as well as data errors. Two of the multiple-error outputs obtained from the syndrome decoder 485 shown in Figure 40 of U.S. Patent Specification No. 4,228,496 are now used, in the present invention, to detect those errors.

The following Table 1 enumerates the 64 possible values of the six-bit syndrome code and gives the interpretation for each possible value as used in the present invention;

TABLE 1

Syndrome Codes

S0	*S1*	*S2*	*S3*	*S4*	*S5*	*Error In*	*S0*	*S1*	*S2*	*S3*	*S4*	*S5*	*Error In*
0	0	0	0	0	0	(No Error)	1	0	0	0	0	0	C0
0	0	0	0	0	1	C5		0		0	0	1	(Double)
	0		0	1	0	C4		0		0	1	0	(Double)
	0		0	1	1	(Double)		0		0	1	1	D8
	0		1	0	0	C3		0		1	0	0	(Double)
	0		1	0	1	(Double)		0		1	0	1	D9
	0		1	1	0	(Double)		0		1	1	0	D10
0	0	0	1	1	1	D0		0		1	1	1	(Double)
0	0	1	0	0	0	C2	1	0	1	0	0	0	(Double)
	0		0	0	1	(Double)		0		0	0	1	D11
	0		0	1	0	(Double)		0		0	1	0	(Multi-All 0's)
	0		0	1	1	(Address)		0		0	1	1	(Double)
	0		1	0	0	(Double)		0		1	0	0	D12
	0		1	0	1	D1		0		1	0	1	(Double)
	0		1	1	0	D2		0		1	1	0	(Double)
	0		1	1	1	(Double)		0		1	1	1	(Multi)
0	1	0	0	0	0	C1	1	1	0	0	0	0	(Double)
	1		0	0	1	(Double)		1		0	0	1	D13
	1		0	1	0	(Double)		1		0	1	0	D14
	1		0	1	1	D3		1		0	1	1	(Double)
	1		1	0	0	(Double)		1		1	0	0	(Address)
	1		1	0	1	(Multi-All 1's)		1		1	0	1	(Double)
	1		1	1	0	D4		1		1	1	0	(Double)
	1		1	1	1	(Double)		1		1	1	1	(Multi)
0	1	1	0	0	0	(Double)	1	1	1	0	0	0	D15
	1		0	0	1	D5		1	1	0	0	1	(Double)
	1		0	1	0	D6		1		0	1	0	(Double)
	1		0	1	1	(Double)		1		0	1	1	(Multi)
	1		1	0	0	D7		1		1	0	0	(Double)
	1		1	0	1	(Double)		1		1	0	1	(Multi)
	1		1	1	0	(Double)		1		1	1	0	(Multi)
	1		1	1	1	(Multi)		1		1	1	1	(Double)

THUS (NUMBER OF 1'S IN SYNDROME)
0 BITS - NO ERROR
1 BIT - CHECK BIT ERROR
2 BITS - DOUBLE
3 BITS - DATA BIT OR MULTI OR ADDRESS
4 BITS - DOUBLE
5 BITS - MULTI
6 BITS - DOUBLE

A data error detection and correction system constructed according to the present invention detects and corrects all single bit data or check bit errors, detects all double bit data or check errors, and detects some data or check errors which involve more than two bits. The present invention also detects all single bit address errors, and detects some mulitple bit address errors, even in the presence of single bit data or check errors. Address errors are not corrected. Operation errors, i.e., differences in states between the operation state machine in the memory modules 21 and the operation state machine in the operation check logic 17, are also detected.

The operation of the memory system 11 described above in the detection and correction of data errors is basically the same as that of the similar structure shown and described in U.S. Patent Specification No. 4,228,496 (see column 72, line 53 through column 78, line 52). The detailed description of the parts of the present invention which correspond to those shown and described in that part of the earlier Specification will therefore not be repeated in this specification. It should be noted that in the present invention the error correction code logic is located in the processor subsystem as shown in Figure 1 as error correction code logic 15, rather than in the memory modules as described in the American Specification.

In particular embodiment of the present invention, the encoding is done in two stages. In a first stage, as illustrated in Figure 5, a 23-bit address is encoded in the 13-bit parity tree 77 and the 10-bit parity tree 75 to produce two parity bits on the lines 81 and 79 respectively. These two parity bits on the lines 81 and 79 are

then encoded according to the connection pattern of the 9-bit trees in the six check bit bus 80, part of the bus 45 shown in Figure 5.

Thus, the 23-bit address is encoded as part of the six check bits in the 22-bit data code word supplied on the bus 45 to the semiconductor storage array 55 (see Figure 2). On a subsequent read operation, a data code word is read from the memory array 55 (see Figure 2) and is transmitted on the read data bus 47 (see Figures 2 and 6) where it is combined with the newly generated read address parity bits transmitted on the lines 79 and 81 to check bit comparator 83 shown in Figure 6. In the check bit comparator 83 these two address parity bits are combined with the data code word read to generate a syndrome on the syndrome bus 85 which indicates the type of error, if any.

An address error and a single data error in combination will always be detected as a double or multiple error.

If an address error alone is detected, it will be reflected as one of two particular codes indicated on the syndrome bus (see Table I above). The detected address error is signalled on line 35 (see Figure 1) from error correction code logic 15 to the memory system error encoder 37, through which the signal passes onto the lines 43 to the CPU 23 and the I/O channel 25. The address error detected is treated as a memory system error by the CPU 23. CPU 23 then uses the information on syndrome bus 85 to determine the type of memory system error that was produced thereon; if the memory system error is an address error, then CPU 23 rejects the data. The CPU 23 then produces an interrupt of the current programme and informs the operator.

Referring to Figures 7 and 8, the operation state machines 61 and 67 assure (with a high degree of fault coverage) that all memory modules 21 and the memory control 13, after every cycle of interaction between the memory subsystem and the processor subsystem, are receiving the same commands and are performing the same operations. The logic function 68 in combination with the state register 70 implement the state machine as defined by the state diagrams 4A - 4F. The state register 70 comprises five flipflops commonly known in the prior art.

Referring again to Figures 3 and 4A - 4F, the state bit on the line 71 and the state bit on the line 52, since they represent in encoded fashion the state of their respective operation state machines, in effect are encoding the sequence of control line states in all previous cyces since the previous reset of the system.

The status of the state machine 67 (see Figure 3) is produced on the line 71. The status of the state machine 61 (see Figure 2) is produced on the line 53. The compare logic 69 (see Figure 3) detects any difference in the state represented on these two lines. This comparison is made only during read cycles. During a read the memory module, selected by the physical address on the bus 33, enables a three-state gate 54 (see Figure 2); otherwise the gate is disabled, i.e., in a high-impedance state. Thus, an operation state bit is applied to the line 53 by the memory module. If a difference in state is detected by the compare logic 69, the compare logic 69 produces a signal on the operation check error line 41 which is supplied to the memory system error encode 37 (see Figure 1), and the memory system error encoder 37, in turn, causes an uncorrectible error signal to be produced on the bus 43. This signal on the bus 43 when received by the CPU 23 then produces a programme interrupt for the operating system programme. The CPU 23 also reads the syndrome code in the syndrome bus 85. The syndrome code will indicate that the error is not a memory data or address error, thus indicating that the error is an operation check in a particular embodiment of the invention disclosed in this application. This signal (the signal on the operation check error line 41) could also be produced independently. In that event, the error transmitted to the CPU 23 would be classified as an operation check interrupt, rather than a general class of uncorrectible error interrupt as described above.

CLAIMS

1. A computer memory control system comprising a processor subsystem and a memory subsystem, the processor subsystem comprising a map/memory control means, error correction code logic means and operation check logic means, and the memory subsystem comprising a memory module having a storage array, bus means connecting the storage array to the map/memory control means and error correction code logic means, timing and control logic means, a control bus connecting the timing and control logic means to the map/memory control means, an operation state bus connected to the operation check logic means, and memory module operation state machine means for generating a signal indicating the status of operations of the memory module and for transmitting that signal to the operation check logic means for comparison to the status of operations of the processor subsystem.

2. A system as claimed in claim 1 wherein the error correction code logic means include a processor subsystem operation state machine means, for generating a signal representing the status of operations of the processor subsystem, and compare logic means, for comparing the signal generated by the memory module operation state machine means and the processor subsystem operation state machine means and for producing an operation check error signal output to the processor subsystem.

3. A system as claimed in claim 1 or 2 wherein the error correction code logic means include address error detection means.

4. A system as claimed in claim 3 wherein the address error detection means include address parity generator means for encoding address parity information into a data check field of each memory location.

5. A system as claimed in any one of claims 1 to 4 wherein the error correction code logic means include syndrome means for identifying address errors as well as data errors indicated in the check field during a

read operation on a data code word read from the storage array.

6. A control module for a computer system processor module of the kind having a map/memory control, error correction code logic and operation check logic, the memory module comprising, a semiconductor storage array, timing and control logic, bus means for connecting the storage array to the map/memory control and error correction code logic and for connecting the timing and control logic to the map/memory control, an operation state bus connectible to the operation check logic of the processor module, and operation state machine means associated with the operation state bus and the timing and control logic for generating a signal indicating the status of operations of the memory module and for transmitting that signal to the operation check logic for comparison to the status of operations of the processor module.

7. A memory module as claimed in claim 6 adapted as an address error apparatus for detecting address errors in the computer system, wherein the error correction code logic encodes a data check field at each memory location for detection of data errors in the data field of each memory location, the module including, encoding means for encoding address parity information into the data check field at each memory location on a write operation, read address means for reading address parity information from the memory array on a subsequent read operation, and comparator means for comparing the parity information encoded during the write operation with the parity information generated during the read operation to generate a syndrome tht indicates the type of address error, if any.

8. A module as claimed in claim 7 wherein the error correction code logic includes parity tree means for parity detection of data errors, and wherein the module is included in the error correction code logic of the computer system as part of the parity tree means.

9. A computer memory control system for protecting data integrity in the main memory of a computer and comprising, a word organized memory with each word location having a data field and a check field, data error detecting means for detecting data and check errors by a syndrome code encoded in the check field of each word, address error detection means for detecting address errors by encoding address parity information into the check field of each word, and operations error check means for detecting any differnce in the sequence of operations between a memory subsystem and a processor subsystem of the computer system.

10. A computer memory control system as claimed in claim 9 wherein the data error detecting means comprise parity tree means for parity detection of data errors.

11. A computer memory control system as claimed in claim 10 wherein the address error detection means use the same parity tree means as the data error detecting means.

12. A computer memory control system as claimed in claim 11 wherein the operations error detecting means comprise a first operation state machine means in the processor subsystem for generating a signal indicating the status of operations in the processor subsystem, an additional operation state machine means in each memory module of the memory subsystem for generating a signal indicating the status of operations in the memory module, and compare logic means for comparing the signal from each memory module with the signal from the processor subsystem and for producing an operation check error signal output to the processor subsystem.

13. A method of detecting address errors in a memory system of a computer of the kind having a word organized storage array with each word comprising a data field and a check field, said method comprising, encoding a data error detecting syndrome code into the check field and encoding address error parity information into the check field so as to detect any address errors by the same syndrome code as used for data error detection.

14. A method of detecting operations errors of a memory module connectible to a computer system processor module, said method comprising, generating at the memory module a signal indicating the status of operations of said module, simultaneously generating at the processor module a signal indicating the status of operations of the processor module, transmitting the status signal of the memory module to the processor module, comparing the two status signals and producing an operation check error signal to the processor module.

15. A computer memory control system substantially as herein described with reference to and as diagrammatically illustated in, the accompanying drawings.

16. A method of controlling a computer memory system substantially as herein described with reference to the accompanying drawings.

Printed for Her Majesty's Stationery Office, by Croydon Printing Company Limited, Croydon, Surrey, 1983.
Published by The Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.